



# Effects of Clustering Algorithms on Typographic Reconstruction

Elisa H. Barney Smith, Bart Lamiroy

## ► To cite this version:

Elisa H. Barney Smith, Bart Lamiroy. Effects of Clustering Algorithms on Typographic Reconstruction. 13th International Conference on Document Analysis and Recognition, Aug 2015, Nancy, France. hal-01154603

**HAL Id: hal-01154603**

**<https://hal.science/hal-01154603>**

Submitted on 22 May 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Effects of Clustering Algorithms on Typographic Reconstruction

Elisa H. Barney Smith

Electrical and Computer Engineering Department  
Boise State University  
Boise, ID 83725-2075  
Email: ebarneysmith@boisestate.edu

Bart Lamiroy

Université de Lorraine – LORIA (UMR 7503)  
Campus Scientifique – BP 239  
54506 Vandœuvre-lès-Nancy CEDEX – FRANCE  
Email: bart.lamiroy@loria.fr

**Abstract**—Type designers and historians studying the typefaces and fonts used in historical documents can usually only rely on available printed material. The initial wooden or metal cast fonts have mostly disappeared. In this paper we address the creation of character templates from printed documents. Images of characters scanned from Renaissance era documents are segmented, then clustered and a template is created from each obtained cluster of similar appearance characters. In order for subsequent typeface analysis tools to operate, the template should reduce the noise present in the individual instances by using information from the set of samples, but the samples must be homogeneous enough to not introduce further noise into the process. This paper evaluates the efficiency of several clustering algorithms and the associated parameters through cluster validity statistics and appearance of the resulting template image. Clustering algorithms that form tight clusters produce templates that highlight details, even though the number of available samples is smaller, while algorithms with larger clusters better capture the global shape of the characters.

## I. INTRODUCTION

Modern typeface design is a contemporary version of a process that has been in use for centuries. Information on how this process has evolved is of interest to type designers and to historians. It enables typographers to precisely study the structures and approaches used since the inception of machine printing. It allows historians to identify and trace similarities, variations and influences of shapes, techniques and models. It allows libraries to provide access to historical scanned documents in a form called *diplomatic translation*, which retains the look of the original, while modernizing its transcription. This increases access to the content, while maintaining a visual experience similar to that with the original document. It can also reduce storage and bandwidth resources.

One of the challenges of typography research is to study the mutual influences and developments in printing techniques and type design, not only in a purely graphical context, but also historical and functional contexts. The net results of the fonts that were physically cast are visible in the books that remain from past eras. The goals that drive this paper are the extraction of information about these fonts from the available printed samples and the retrieval the underlying typeface parameters. We desire to extract information from images of the characters on printed pages of Renaissance era books that can be used to

express those typefaces in numerical description languages as METAFONT or UFO.

Our main idea is to create a high resolution “ideal shape” or exemplar for each character in the printed text. To achieve this, pages are scanned, and segmented into individual character instances. The characters are grouped by a glyph clustering algorithm and are manually labeled with their character class or lexicographic meaning. For each glyph class an exemplar is computed and then vectorized (*cf.* [6] for a first approach).

The document analysis community [4, ch. 30] has developed the basic techniques to accomplish this task. However, most of these digital tools, and the state-of-the-art in general, take the approach of considering the variability and characteristics of forms like “noise” that needs to be filtered to achieve the recognition of letters. Our concern is rather different: in this paper the actual *recognition* of the characters is secondary to the extraction of the typographic information. It is therefore necessary to precisely detect and encode the characteristics of the typefaces to be able to reconstruct accurate representations. The choices of how to configure these common tools to accomplish this goal are many. The ones explored in this paper are (1) the choice of clustering algorithm for identifying glyphs, (2) the features to be used in those algorithms (3) the distance metric to determine when similarity between glyph images exist and (4) how to reconstruct the prototype once several samples are accurately identified.

The subsequent sections of this paper will provide an overview of related work and describe the various steps of our approach.

## II. BACKGROUND AND PAST WORK

Somewhat surprisingly, the study of typography and analysis of specific graphic forms of writings is absent from the state-of-the-art in Document Image Analysis<sup>1</sup>. In general, the work and research is confined to content analysis and syntactic restitution of the content (text transcription, filtering or cleaning artefacts affecting reading, looking for repetitive patterns, *etc.*). Rare works integrating typographic information such as [1], [7]–[9] do so in the sole aim of improving OCR type transcription engines or to classify similar typefaces,

---

Prof. Barney Smith gratefully acknowledges funding through the *Chercheur d'Excellence* program by the *Conseil Régional de Lorraine* in 2015.

---

<sup>1</sup>There exists a rich state-of-the-art reporting shape analysis proper to type design and typography, but it is either related to capturing the design process itself, or falls into the domains of Comparative Art or History of Arts, and therefore doesn't focus on the Pattern Recognition and Image Analysis parts.

and are not specifically looking to characterize the fonts for subsequent reuse.

This reuse is nevertheless relevant in certain contexts. In [13], for example, Marc Smith identifies several levels and modalities of the transcription of ancient texts: translation (modernized glossary), modernized transcription (standardized spelling), conservative transcription, imitative diplomatic transcription or *facsimile*. These different levels, on an axis from the interpretive transcription (focusing on the content) to the imitative transcription (favoring the form) are cases where typefaces need to closely reproduce the forms of the original document. The state-of-the-art document image analysis is totally absent from this area. To provide some insights into how it can contribute to models or technological tools addressing issues of typographic analysis it is necessary to revisit the image analysis approaches used in other contexts [4] to study their relevance and “applicability” therein.

In this paper image scanning and analysis technologies are used to extract and represent typographical features of Renaissance documents to facilitate typographic study. We are considering the following processing workflow [6]:

- 1) establish a baseline corpus by a series of scans of characters from identified printed Renaissance sources (pages of the same book, same typesetter ...);
- 2) develop image processing and classification algorithms to identify the best exemplars of all the glyphs;
- 3) detect edges and skeletons and subsequently vectorize the alphabet.

This paper will focus only on the second stage only. In order for subsequent typeface analysis tools to operate, the extracted exemplars should reduce the noise present in the individual instances. Therefore, the samples must be homogeneous enough to not introduce further perturbation. We will evaluate the efficiency of several clustering algorithms and their associated parameters through cluster validity statistics and appearance of the resulting template image.

### III. TECHNICAL BACKGROUND

As mentioned before, the specific goal in this paper is to create a representative exemplar for each character class that would have a sufficiently high level of precision to allow further vectorization. The approach used in prior work [6] was to segment individual characters from the digitized document [10], and then apply clustering to the resulting samples [11]. Throughout this paper, we will be reusing the segmentation results from [6] that were obtained with the AGORA tool<sup>2</sup> for which professional typographers have empirically set the parameters to extract the characters.

The RETRO software uses a single exemplar agglomerative clustering algorithm. Although the supporting published work [11] never explicitly mentions the clustering method that is used, reverse engineering from the software’s implemented code clearly establishes its characteristics. This agglomerative clustering method takes the first instance of the list of considered items and elects it as the exemplar for the

first cluster. Then, every subsequent character is compared with all the existing cluster exemplars. If its distance to the closest exemplar is smaller than a set threshold (selected to be 125 as in RETRO), then the character is added to that cluster. Otherwise it forms a new cluster and becomes the exemplar for that cluster. Exemplars do not change during the process. For comparison and experimental validation, we have reimplemented this clustering algorithm.

The rest of the paper will look into how the choice of clustering parameters influences subsequent segmentation of typography features. The parameters under consideration are: (1) the clustering algorithm itself; this paper compares agglomerative hierarchical clustering and RETRO’s single exemplar agglomerative clustering, (2) the cluster distance metrics, (3) the cluster validity metrics, (4) the choice of features.

#### A. Cluster Distance Metrics

Agglomerative hierarchical clustering initializes with the number of clusters equal to the number of data items: each character belongs to its own cluster. Then, iteratively, all clusters are compared to find the pair for which a given set distance is minimal. Those two clusters,  $X_i$  and  $X_j$  are then merged into a new cluster, and the process is repeated until a specific threshold is reached (this threshold may either be a pre-determined number of clusters, or specific conditions on the cluster properties).

The set distance metric used to compare two clusters is one of the major operational parameters of this algorithm. In our experiments, we will be comparing the following ones:

$$D_{min}(X_i, X_j) = \min_{x \in X_i, x' \in X_j} \|x - x'\|, \quad (1)$$

$$D_{max}(X_i, X_j) = \max_{x \in X_i, x' \in X_j} \|x - x'\|, \quad (2)$$

and

$$D_{avg}(X_i, X_j) = \frac{1}{N_i N_j} \sum_{x \in X_i} \sum_{x' \in X_j} \|x - x'\|. \quad (3)$$

$D_{min}$  determines the minimum distance between any pair of points between the two sets,  $D_{max}$  determines the maximum distance between any pair of points between the two sets, and  $D_{avg}$  determines the average distance between any pair of points between the two sets.

#### B. Cluster Validity Metrics

Every clustering algorithm will divide the data into the number of clusters specified. To aid in determining whether these are good clusters or not, several cluster validity metrics have been proposed in the literature. They generally express whether the distance between members within a cluster is small, and the distance between clusters is large. Two common ones are used in this paper: Dunn [5], and Davies-Bouldin [2].

- Cluster validity is defined by Dunn [5] as

$$V_D(u) = \min_{i \in 1..C} \left\{ \min_{\substack{j \in 1..C \\ j \neq i}} \frac{\delta(X_i, X_j)}{\max_{k \in 1..C} \Delta(X_k)} \right\}. \quad (4)$$

<sup>2</sup>The software is freely available from <https://sites.google.com/site/paradiitproject/results>

- The Davies-Bouldin Index [2] is calculated by

$$V_{DB}(u) = \frac{1}{C} \sum_{i=1}^C \max_{\substack{j \in 1..C \\ j \neq i}} \left\{ \frac{\Delta(X_i) \Delta(X_j)}{\delta(X_i, X_j)} \right\}. \quad (5)$$

$X_i$  and  $X_j$  are clusters, and  $C$  is the total number of clusters. In both metrics,  $\delta(X_i, X_j)$  is the intercluster distance. This can be any distance that measures the distance between clusters. We used minimum distance between furthest cluster points.  $\Delta(X_k)$  is the intracluster distance, the distance within the cluster. We used the maximum distance between any two cluster member points. A larger value of  $V_D$  indicates the clusters are good clusters, whereas a smaller value of  $V_{DB}$  indicates the clusters are good clusters.

### C. Choice of Features and Distances

There are many choices of features to use when classifying or clustering shapes, and there are many distance metrics that can be used with them. As mentioned before, the general features developed in the state-of-the-art that are most popular, focus on recognition, aim to discard small differences in samples, and highlight the major shape characteristics. For this project we want to maintain some level of the intra-character class details, so choosing features that will form compact clusters is important.

We are running only on single pages of documents. Our dataset therefore has no relative skew, and because it was printed by type that theoretically is identical in size, there is no scale difference to compensate for. Features we are using are pixel features in bilevel, gray scale and color. In this paper we will consider 3 types of pixel features: bitonal, grey level and color. Future work will extend the scope to a wider range of features such as moment features and shape descriptors. Several possible distance metrics exist for  $\|x - x'\|$ . We consider three distance metrics: Hamming,  $L_1$  and  $L_2$ .

The difficulty with these features and metrics is that they require images to be well aligned. We consider three basic methods to align the characters: using (1) the bounding box of the character (as done by RETRO [11]), (2) the calculated center of mass (COM) of the character, or (3) through finding the relative phase shift that produces the maximum correlation (X-CORR) between two characters over a  $\pm 5$  pixel search range around the COM. It is to be noted that the bounding boxes are highly dependent on the quality of the segmentation and often introduce instability. We keep them as a reference for comparison with RETRO [11].

### D. Overview of Operational Parameters

In the light of the previous sections, there are several operational parameters that define our approach. These parameters are:

- 1) 4 different clustering approaches: RETRO's single exemplar agglomerative clustering (S-AGGLOM), plus agglomerative hierarchical clustering with the three distance metrics:  $D_{min}$ ,  $D_{max}$  and  $D_{avg}$ ;
- 2) each of these cluster metrics depends on the choice of features. We have selected 3 different features: bitonal, grey level and color image pixels.

- 3) the feature depends on the accompanying distance metric  $\|x - x'\|$ . We use Hamming,  $L_1$  and  $L_2$ .
- 4) Furthermore, the pixel features require an image alignment for which we have identified 3 possible algorithms: bounding-box center alignment, center of mass alignment (COM) and maximum correlation alignment (X-CORR);
- 5) the evaluation metric. The best clustering partition can be decided either using Dunn or Davies-Bouldin, both of which depend on  $\delta(X_i, X_j)$  and  $\Delta(X_k)$ ; the two latter have been fixed to respectively minimum distance between opposite cluster points the maximum distance between any two cluster member points. In our case, we also use reprojection error after vectorization.

It should be clear to the reader that many other choices are possible for each of these operational parameter categories, as we have already pointed out earlier. However, the choices retained in this paper alone already give rise to approximately 100 different clustering implementations that, as we shall show, may yield significantly different results.

## IV. EXPERIMENTAL PROTOCOL

As mentioned in the introduction, this work is part of a larger initiative to try and use document image analysis to recover typographic parameters from scanned instances of printed text. Ultimately this should result in reverse engineering images into complete digital font descriptions. This paper addresses the issue of measuring how the choice of clustering algorithms influences the quality of the achieved font description.

Our experimental protocol is as follows:

- 1) We use a scanned page of a XIV<sup>th</sup> century edition of Montaigne's "Essays" [3] from which we have extracted 2951 individual glyphs using AGORA<sup>3</sup>. The scan is of intermediate resolution ( $2334 \times 3802$  pixels for an *in folio*, i.e. approx. 300 dpi). Some of the extracted characters are shown in Figure 1.
- 2) We apply all realizations of the family of clustering algorithms described previously.
- 3) We have selected a number of reasonable conditions and measurements reducing the search space. These will be described below.
- 4) For each obtained clustering algorithm we compute the representative exemplar for this cluster (cf. Section IV-B). This exemplar is then converted into a digital vector shape using Potrace [12].
- 5) Finding a reliable quality metric to assess the value of the obtained vector shape description is currently work in progress, since it relies on semi-subjective interpretations, based on type designers' domain knowledge. We have therefore opted for a more neutral approach: the vector shape is reprojected into image space, and we compute the distance from the cluster exemplar. Both the cluster exemplar and reprojected vector shape are also compared to the rest of the cluster members to get the average, median, min and max distances.

<sup>3</sup>The raw data is available from <http://re-typograph.loria.fr>.

### A. Distance Metric, Feature Type and Alignment Method

The difference measured between two characters depends on the distance metric used, the feature that is being measured, and since the features are pixels, how the pairs of characters are aligned. The characters can be represented with bilevel, gray scale or color pixels. Four distance metrics were considered: Hamming, gray level  $L_1$  and  $L_2$ , and color  $L_2$ . Their optimal use requires the characters to be pre-aligned (see above). To choose these parameters, we used confusion cases of letters such as c/e, and t/r (which for this font are very similar) to produce a Receiver Operating Characteristics (ROC) curve. We calculated the difference between pairs of characters between and within classes and measured the percentage of correctly and falsely classified characters across a range of thresholds. Due to space constraints this paper does not include the details of the results of these experiments. Because they performed best, we shall be using X-CORR on grey level images using  $L_2$  distances through the remainder of this paper.

### B. Forming the Exemplar Image

The representative image for a cluster is formed by determining the center points for the characters based on the methods described for aligning the character samples for distance calculation for clustering. Those character center points are then used to build a (possibly) new bounding box. All these bounding boxes are made the same size, based on the maximum size of all characters to be combined.

Three ways of combining these characters were considered. The first averages the bilevel images, produced with an empirically chosen threshold of 150. The second averages the gray level images, and the third averages all three color planes of the color image.

## V. EXPERIMENTS AND RESULTS

The 2951 characters on the first page of [3] were clustered with each of the four clustering algorithms under consideration. Approaches to evaluating the results took two basic forms: (1) analysis of the clusters and (2) analysis of the resulting exemplar. The results are described next and show that both analyses come to the same conclusion.

### A. Effect of Clustering

We explored the choice of clustering algorithms and their parameters. The overview of the results is shown in Table I. Dunn and Davis-Bouldin show a preference for respectively  $D_{max}$  and  $D_{avg}$  (the other coming second, for both measures).

TABLE I. CLUSTERING RESULTS, N=2951 POINTS, DISTANCE THRESHOLD = 2

Clustering method	Number of clusters	Max cluster size	Dunn Validity metric	Davies-Bouldin Validity metric
$D_{min}$	118	852	0.0027	2.4780
$D_{max}$	236	28	<b>0.2588</b>	0.1121
$D_{avg}$	189	166	0.1327	<b>0.0928</b>
S-AGGLOM	150	852	0.0007	1.8743

The fewest (and largest) clusters are formed by  $D_{min}$ . They are very elongated. One character can be similar to another, which is in turn similar to a third. The characters at opposite ends of a chain may not be very similar to each other. On

the other end of the spectrum  $D_{max}$  produces many small, tight clusters. For OCR type applications this is not appealing, but for this work it may be a desirable trait, so long as each cluster contains “enough” characters to form a representative character for each character class.  $D_{avg}$  produces results in between  $D_{min}$  and  $D_{max}$ . This has the advantage of producing larger clusters, so more exemplars are available to make an “average” character. Single exemplar agglomerative clustering only compares each character once to each cluster’s exemplar so since the exemplar on which a cluster is based is chosen from the first sample to enter a cluster, it is very dependent on the order in which data is presented to it. It therefore can have members that are similar to the anchor point, but are not similar to each other. Unlike the  $D_{min}$ , all distances are to a single anchor exemplar.

Since the final goal of this work is to provide a first stage input for further extraction of high level shape information, we examine the quality of the exemplars created for each cluster.

### B. Evaluation of Exemplars

All clusters with more than one member were manually labeled. For the sake of clarity, we will be focusing only on the exemplars for those labeled with the letters ‘d’ or ‘e’. These letters were chosen because the letter ‘d’ was studied in [6], and because the letter ‘e’ occurs so frequently that it appears well formed, as well as degraded in many different manners. It will have enough samples to evaluate the results when it remains as one cluster and when it divides.

The representativeness of the exemplar of a cluster can be related to its distance from all its cluster’s members. Table II shows the statistics about these distances calculated for these letters with bilevel templates. The number of clusters and the size of the clusters are as in Table I. The total number of characters on the page appears to vary by clustering method. This is because only clusters that had more than one point were considered for this analysis. The clusters formed with  $D_{min}$  and S-AGGLOM are very large, and thus the average distance from the exemplar to the points is larger. The clusters formed with  $D_{max}$  and  $D_{avg}$  are smaller, so their distances are smaller. This is in line with the Dunn and Davis-Bouldin metrics above.

TABLE II. EXEMPLAR DISTANCE STATISTICS, N=2951 POINTS, DISTANCE THRESHOLD = 2

Clustering method		Num Points	Average dist.	Std. dev.	Median dist.	Min dist.	Max dist.
$D_{min}$	d	93	0.177	0.022	0.178	0.128	0.23
$D_{min}$	e	534	0.136	0.024	0.137	0.056	0.21
$D_{max}$	d	93	<b>0.078</b>	0.014	0.078	0.061	0.10
$D_{max}$	e	402	<b>0.079</b>	0.014	0.079	0.064	0.09
$D_{avg}$	d	89	0.100	0.020	0.100	0.077	0.13
$D_{avg}$	e	399	0.086	0.016	0.086	0.069	0.10
S-AGGLOM	d	91	0.171	0.022	0.172	0.121	0.22
S-AGGLOM	e	420	0.098	0.019	0.097	0.073	0.12

Figure 2 shows instances of the exemplars created by averaging cluster results, and processing with Potrace. They share some characteristics, but also have many differences. For instance, the lower serif in the exemplar (a) created with  $D_{min}$  is more rounded than the sample in (b). This can be explained by the higher number of points used in averaging. The samples in subfigures (b) and (c) were both made with

4 samples from  $D_{max}$  clusters and significance difference in shapes can be seen. This can not be explained by an increased amount of averaging, because the number of points used is the same. The  $D_{max}$  clustering forcing the characters to be more homogeneous enables the differences in the characters across the page to be seen. It is possible that the characters came from a mixed case of type. It is also possible that there are other printing and imaging defects involved. The important thing to note is that these differences are only available because of the tight clustering resulting from using the  $D_{max}$  set distance.

The exact impact of the above points will need to be studied in depth in tight collaboration with expert type designers, in order to focus on the appropriate visual characteristics. This is beyond the scope of this paper. Within the scope of this paper, however, is the observation of the effect of the clustering on the final vectorization. We measured this by reproducing the same measurements as those of Table II but instead of using the cluster exemplar, we use the Potrace output. The results can be found in Table III.

TABLE III. POTRACE DISTANCE STATISTICS UNDER THE SAME CONDITIONS AS TABLE II

Clustering method		Num Points	Average dist.	Std. dev.	Median dist.	Min dist.	Max dist.
$D_{min}$	d	93	<b>0.050</b>	0.011	0.048	0.027	0.096
$D_{min}$	e	534	<b>0.047</b>	0.015	0.045	0.021	0.136
$D_{max}$	d	93	0.054	0.010	0.054	0.042	0.066
$D_{max}$	e	402	0.078	0.011	0.077	0.066	0.091
$D_{avg}$	d	89	0.058	0.013	0.057	0.042	0.076
$D_{avg}$	e	399	0.080	0.013	0.079	0.066	0.095
S-AGGLOM	d	91	0.052	0.012	0.051	0.028	0.099
S-AGGLOM	e	420	0.078	0.013	0.076	0.063	0.100

We observe two major effects. First, there is a significant decrease in average distance and standard deviation, compared to the computed exemplar. This means that the computation of the exemplar is sub-optimal, but that the vectorization step is capable of filtering some of the artefacts (like the blurring, for instance). This seems to be confirmed by the second effect: the difference is greater for  $D_{min}$  and S-AGGLOM, who carried larger clusters, and therefore introduced a larger blurring effect.

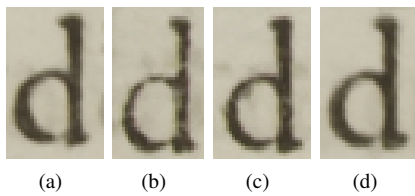


Fig. 1. Samples of individual occurrences of a 'd' output from AGORA used in this work. Note the variability in alignment, quality and detail.

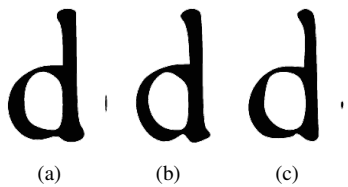


Fig. 2. Comparison of template outputs. (a) created from 93 samples from  $D_{min}$  clustering. (b,c) created from 4 samples of  $D_{max}$  clustering.

## VI. CONCLUSION

The  $D_{max}$  hierarchical clustering algorithm provides the most sensitivity to the variations in the characters. It produces many small clusters and is therefore the ideal choice for applications looking to detect subtle changes in shape (for instance, cast font defects).  $D_{min}$ , on the other hand, is more likely to capture general shape and form the basis for overall typeface characterization. It fails to capture the finer details of the shape, however (e.g. the curves and extremities of the serifs in Figure 2). We know that these details are of particular importance for the characterization of the typeface.

The exhaustive nature of the parametrization of clustering approaches developed in the paper forms a solid and well laid-out basis for further investigation into the specific shape characteristics type designers need to re-engineer type faces. We are currently working with them, based on the results provided in this study, in order to capture those characteristics in more detail and to refine both the clustering parameters and the subsequent vectorization algorithms.

## REFERENCES

- [1] F. Bapst and R. Ingold. Using typography in document image analysis. In *Proc. Electronic Publishing, Artistic Imaging, and Digital Typography, 7th Int. Conf. on Electronic Publishing, EP '98*, pages 240–251, St. Malo, France, April 1998.
- [2] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2):224–227, February 1979.
- [3] M. de Montaigne. *Les Essais*. A Paris, Chez Pierre Rocolet, Imp. & Libraire ordinaire du Roy, au Palais, en la des prisonniers, Paris, 15 juin 1635, chez Toussaint du Bray, rue Saint Jacques, aux Espies-meurs et Pierre Rocolet edition, 1580.
- [4] D. Doermann and K. Tombre, editors. *Handbook of Document Image Processing and Recognition*. Springer London, 2014.
- [5] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57, January 1973.
- [6] B. Lamiroy, T. Bouville, J. Bléjean, H. Cao, S. Ghamizi, R. Houpin, and M. Lloyd. Re-Typograph Phase I: a Proof-of-Concept for Typeface Parameter Extraction from Historical Documents. In *Document Recognition and Retrieval XXII*, volume 9402, San Francisco, United States, February 2015.
- [7] F. LeBourgeois and H. Emptoz. Document analysis in gray level and typography extraction using character pattern redundancies. In *Proc. of the 5th Int. Conf. on Document Analysis and Recognition, 1999. ICDAR '99*, pages 177–180, Sep 1999.
- [8] R. LiVolsi, R. Zanibbi, and C. A. Bigelow. Collecting historical font metrics from Google books. In *Proc. of the 21st Int. Conf. on Pattern Recognition, ICPR*, pages 351–355, Tsukuba, Japan, November 2012.
- [9] J. Lorenz and M. Weskamp. Generating typefaces through image analysis, December 11 2012. US Patent 8,331,681.
- [10] J.-Y. Ramel, S. Leriche, M.L. Demonet, and S. Busson. User-driven page layout analysis of historical printed books. *Int. Journal of Document Analysis and Recognition (IJAR)*, 9(2-4):243–261, 2007.
- [11] J.-Y. Ramel, N. Sidère, and F. Rayar. Interactive layout analysis, content extraction and transcription of historical printed books using Pattern Redundancy Analysis. *Literary and Linguistic Computing*, 28(2):303–314, January 2013.
- [12] P. Selinger. Potrace: a polygon-based tracing algorithm, 2003. Potrace (online), <http://potrace.sourceforge.net/potrace.pdf> (2009-07-01).
- [13] M. Smith. La typographie face aux écritures anciennes : entre reproduction et transcoding. Introductory talk at the colloque international Gestion informatisée des écritures anciennes, Université de Tours, Centre d'études supérieures de la Renaissance, May 2013.